Which One? Grounding the Referent Based on Efficient Human-Robot Interaction

Which One? Grounding the Referent Based on Efficient Human-Robot Interaction

Raquel Ros[1,2], Séverin Lemaignan[1,2], E. Akin Sisbot[1,2], Rachid Alami[1,2],
Jasmin Steinwender[3], Katharina Hamann[3], Felix Warneken[4]

[1]CNRS - LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
[2]Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France
[3]Dept. Developmental and Comparative Psychology, Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany
[4]Dept. Psychology, Harvard University, Cambridge, MA, USA
Email: {rrosespi, slemaign, easisbot, rachid}@laas.fr, {jasmin_steinwender,khamann}@eva.mpg.de, warneken@wjh.harvard.edu

***Abstract*— In human-robot interaction, a robot must be prepared to handle possible ambiguities generated by a human partner. In this work we propose a set of strategies that allow a robot to identify the referent when the human partner refers to an object giving incomplete information, i.e. an ambiguous description. Moreover, we propose the use of an ontology to store and reason on the robot's knowledge to ease clarification, and therefore, improve interaction. We validate our work through both simulation and two real robotic platforms performing two tasks: a daily-life situation and a game.**

I. INTRODUCTION

In daily human interactions, where people refer to objects ("Look at the bike"), sometimes the utterance does not contain sufficient information to be understood correctly. That is, ambiguities concerning the referent can occur ("Which of the two bikes visible to me does she mean?"). To establish an efficient exchange of information and thus communicate meaning, these ambiguities have to be resolved. Humans employ several basic strategies in order to clarify such ambiguities, and they do so efficiently and smoothly. First, by applying internal cognitive strategies; and only later, when those proved unsuccessful, verbal inquiries come into play.

In human-robot interaction, the robot must be prepared to handle possible ambiguities generated by the human partner. We believe that a robotic system should include a clarification strategy which allows to find the referent autonomously if possible, for two reasons. First, humans are not always aware of when they create ambiguities and therefore, they will expect that the robot will be able to resolve them internally. And second, a robot that is not able to resolve ambiguities by itself would have to constantly inquire the human for clarification which would result in a tedious human-robot interaction.

One disambiguation strategy applied by humans is to take the other's visual perspective into consideration. Studies in developmental psychology demonstrate that even very young children rely on the speaker's visual access. They take into account that others might see things they themselves do not see [1]. When two objects are available to the child and the adult can only see one of them, they understand that whenever the adult is looking for an object, she can only refer to the one she cannot see [2]. This ability to engage in visual perspective taking is thus one fundamental strategy in solving referential ambiguities among humans.

In the field of human-robot interaction, few work has been done on applying perspective-taking mechanisms for ambiguity resolution. Trafton et al. proposed in [3], [4] a system by which the robot is able to figure out which of several cones the human is referring to in different situations (visible/not visible for one of the interacting agents). Berlin et al. [5] focused on the use of visual perspective taking skills for learning from a human teacher. Visual perspective taking has been also used to aid action recognition between two robots [6]. In the present work, we propose the use of visual perspective taking based on [7] to ease clarification of referential utterances in scenarios with multiple objects.

However, we believe that apart from this fundamental cognitive mechanism of visual perspective taking, the robot should be provided with means to extract and clarify as much information from both the environment and the speaker's utterance as possible. For instance, the speaker might refer to a specific frame of reference, which can be observer-relative (as in the case of referring to an object "on the right") and thereby producing ambiguities comparable to the ones just discussed. Alternatively, the speaker might specify the location of the desired object in relation to another object ("X is inside of Y"). Finally, he might provide information about specific features like the object's size or color. This work, therefore, is additionally directed at supplementing the robot with mechanisms covering such demands.

In particular, we introduce an approach for finding the referent based on a set of descriptors when incomplete information has been provided (incomplete in the sense of ambiguous information, i.e. more than one object fulfills the given description). To this end, the robot should reason based

This research was supported by a Marie Curie Intra European Fellowship and the European Community's Information and Communication Technologies within the 7th European Community Framework Programme under grant agreements no. [220368], ARBI and [215805] CHRIS projects.

on its current knowledge about the world and interact with its human partner, if required, in order to obtain further information that will allow the robot to identify the referred object. The robot's current knowledge of the world is based on the description of objects in the environment and assumptions about the visual access of its human partner. A first attempt was introduced in [8]. The novelty of the work here is that the robot's knowledge is represented by an ontology. The advantage of using an ontology is that it is not only used as a central knowledge repository, but more importantly, that it provides a semantic level allowing a certain degree of reasoning on the stored knowledge. To validate our approach, we present two types of tasks: a daily situation where a human ask for an object using ambiguous information, and a game, which exploits the reasoning ability of the robot through the use of the ontology. We must remark that we do not aim at dealing with natural language understanding or conversational reasoning. Our work is mainly centered on finding the "right" discriminants to ground the referent.

The paper is organized as follows. Section II goes through the different types of information that compose the knowledge of the robot used in this work. Next, in Section III we describe our ontology-based approach for finding out the referent. Integration and validations scenarios are detailed in Section IV and finally, conclusions and future work are presented in Section V.

II. The Robot's Knowledge

In this section we describe the different sources of information that take part of the robot's knowledge (about the world and the agents in) used in this work. This information is then used to disambiguate between different objects.

A. Visual Perspective Taking

Visual perspective taking refers to the ability for visually perceiving the environment from other's point of view. This ability allows us to identify the referent in situations when the visual perception for one person differs from the other one. In developmental psychology, one typical example consists of two similar objects in a room (eg. two balls) where both are visible for the child, but only one is visible for the adult. Thus, when the adult asks the child to hand over "the ball", the child is able to correctly identify which ball the adult is referring to (i.e. the visible one from the adult point of view), without asking.

The robot computes the visibility information for each object (or agent) in the environment with respect to each agent. This information is stored in each agent's cognitive model (we will come back to this aspect in Section III-A).

In [7], [8] we present a model-based approach for implementing visual perspective taking abilities. In this approach, 2D perspective projections of the 3D environment (Figure 1a,b) is used to determine if an object is visible to an agent. We first obtain the projection of the isolated object (Figure 1c, the blue box), and we compare it with the "real" projection of the scene which considers occlusions of the evaluated object (Figure 1d, the teddy bear is partially occluding the blue box). A visibility ratio of the object is then computed by comparing both images. An object is visible to an agent if the ratio is over a given threshold.

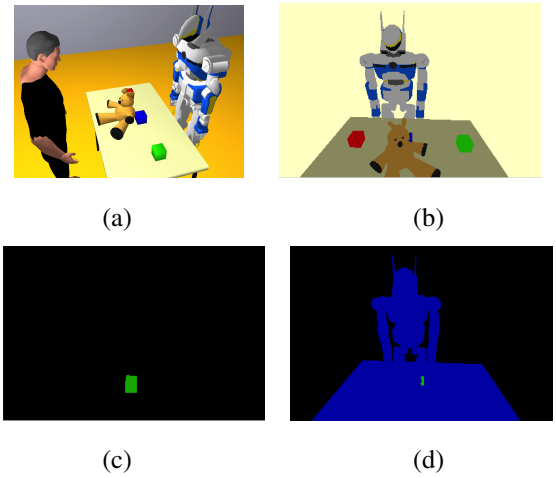

Fig. 1. (a) An example of the environment, (b) human visual perspective, (c) free relative projection and (d) visible relative projection.

In order to obtain a visual perspective, the actual visibility alone is not enough. We believe that visual perspective taking ability is not restricted to what the other person is seeing in a given moment, but also what he "can" see with a minimal effort (moving the eyes or the head). To model the potential visibility of an object we compute the visibility ratio while turning the head of the agent model towards the object.

In order to enrich the visual perspective model and reason on the human's focus of attention, the placement of the object respect to the human's vision is also computed. According to human's gaze direction and object's position, we compute whether the object is within the human's focus of attention (FOA), field of view (FOV) or out of field of view (OOF).

B. Spatial Perspective Taking

Spatial perspective taking refers to the qualitative spatial location of objects (or agents) with respect to a frame (eg. the keys on my left). Based on the frame of reference, the description of an object varies. Humans mix perspectives frequently during interaction. This is more effective than maintaining a consistent one, either because the (cognitive) cost of switching is lower than remaining with the same perspective, or if the cost is about the same, because the spatial situation may be easily described from one perspective rather than another [9]. Ambiguities arise when one speaker refers to an object within a reference system (or changes the reference system, i.e., switches perspective) without informing her partner about it. For example, the speaker could ask for the "keys on the left". Since no reference system has been given, the listener would not know where exactly to look. However, asking for "the keys on your left" gives enough information to the listener to understand where the speaker is referring to. The reference system has to be defined properly because the terms of reference (left, right, above,...) may be identical in different systems [10].

On the contrary, when using an exact, unambiguous term of reference to describe a location (eg. "go north") no ambiguity arises.

In this work, we use two types of the frames of reference: egocentric (from the robot perspective) and addressee-centered (from the human perspective). Thus, given an object and the referent we divide the space around the referent into four regions: front, left, right and back. The number of these regions are doubled with the distinction of near and far from the referent in the center. These regions are separated by arbitrary angle values relative to the referent orientation. Depending of the task the number of regions can be increased to 16 to include a more precise spatial placement information (e.g. "near front right", "far back left").

C. Symbolic Location Descriptors

Symbolic location descriptors allow the robot to represent spatial relations between objects in the environment. They are computed based on the 3D geometric world representation. In this work we propose the use of three basic symbolic relations between each pair of objects. However, their inverse relations can be automatically computed enlarging the symbolic descriptions easily:

IsIn: indicates if an object (or an agent) is inside of another object. It is computed by testing if the bounding box of an object is "completely inside" the bounding box of another object. Eg. `Bottle IsIn TrashBin`. Its inverse relation corresponds to `TrashBin Contains Bottle`.

IsOn: indicates if an object (or an agent) is placed on top of another object. We test if the bottom of the object's bounding box is placed higher than the top of another object's bounding box and lower than an arbitrary value (this value depends on the errors of the object perception system). Eg. `Red-box IsOn Blue-box`. Its inverse relation corresponds to `Blue-box IsUnder Red-box`.

IsNextTo: tests if an object (or an agent) is next to another object. It is based on the distance between two objects and their relative placement: if neither one of the objects is placed higher or lower wrt the other, and neither one is inside the other, and if the distance between both objects is not greater than the longest dimension of the bigger object, then the objects are considered next to each other. Eg. `Bottle IsNextTo Cup`. There is no inverse relation, but symmetric, i.e. `Cup IsNextTo Bottle`.

D. Feature Descriptors

Objects have features (like color, size, shape, texture, etc.) that allow us to distinguish one from another. Besides, we can also categorize objects in different classes and refer to their class as a descriptor. For example, a glass is an object that can be classified based on its purpose in different ways, such as a beer glass, a wine glass, a champagne glass, and others. However, the wine glass can also be subdivided in two categories, white wine glass and red wine glass. Hence, in a scenario with three glasses (a champagne glass, a white wine glass and a red wine glass), simply asking for "the glass" would bring out ambiguities. Asking for "the wine glass", still would produce confusion. The only unambiguous feature description would be asking for "the red wine glass" instead.

In the current approach, the robot cannot perceive these type of features by itself (due to limitations in perception, which is not the focus of our work). Thus, we have to explicitly inform them to the robot. So far, this information is loaded into the ontology during initialization.

III. Ontology-Based Clarification Process

Given a complete or incomplete statement, the goal is to determine the referent based on the current knowledge of the robot. In this section we first introduce the ontology used in our robot for storing and reasoning on its knowledge, and then the ontology-based approach for resolving the referent.

A. ORO - The Ontology

ORO (the "OpenRobot Ontology" server [11]) is a central knowledge repository that stores, manages, processes and exposes knowledge for the robot. It formally represents statements on the world as triples `<subject> <predicate> <object>`. It uses two open-source libraries: Jena for storage and manipulation of statements and Pellet first-order logic reasoner to classify, apply rules and compute inferences on the knowledge base.

ORO defines an initial *upper* ontology for human-aware robotics called *OpenRobots Commonsense Ontology*. This initial ontology contains a set of concepts, relationships between concepts and rules that defines the "cultural background" of the robot, i.e. the a priori known concepts. Currently, this commonsense knowledge is focused on the requirement of human-robot interactions in everyday environments, but contains as well generic concepts like `thing`, `object`, `location` and relationships between those.

Besides simply storing and reasoning about knowledge, ORO offers several useful features for human-robot interaction. One advantage offered by the ORO architecture is that independent cognitive models for each agent can be maintained. When the robot interacts with a new agent, a separate triple storage is created to store the robot's knowledge about the agent's perception. For instance, in the case of perspective taking, we compute the visibility and spatial information about the world from each agent point of view, and store it in their own cognitive models. Having separate cognitive models allows us to store and reason on different models of the world.

Regarding the ontology performance, it has proved to be fast enough to solve the problems presented in this work. In a benchmark example, it is able to solve around 73000 simple queries per second and 7000 more complex ones per second.

B. Clarification Algorithm

The ontology is first initialized with the description of the environment represented by object features as defined in Section II-D which is considered the robot's initial knowledge about the world (along with the common sense concepts).

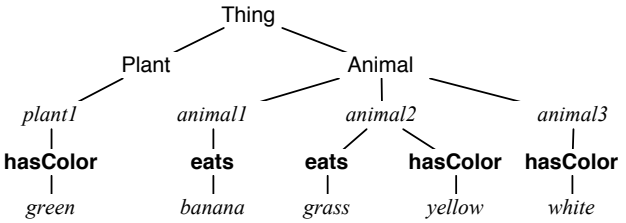

Fig. 2. Ontology example. Names with first capital letter correspond to classes (type); bold names, to properties; and italic names, to instances.

During interaction, the robot's knowledge is updated with the incoming information from the geometric reasoning, i.e. visual perspective taking, spatial perspective taking and symbolic locations descriptors. Based on all this information, and a given partial (or complete) description of an object (list of attribute-value pairs), the robot is able to identify the referred object the following way (Algorithm 1). First it obtains all objects that fulfill the initial description. Based on the result it either succeeds (obtains one single object), fails (no object with that description could be found) or obtains several objects. In this latter case, a new descriptor is added to the initial description and the process starts over again. Failure occur when the description does not match any object from the robot's knowledge. Either because the robot's knowledge is incomplete (the human refers to an unknown descriptor or descriptor value) or due to inconsistent information (human's and robot's beliefs differ).

Algorithm 1 clarify($description$)

1: $objectL \leftarrow$ get_obj_with_desc($description$)
2: **if** $length(objectL) == 1$ **then**
3: **return** $first(objectL)$
4: **else if** $length(objectL) == 0$ **then**
5: **return** no_object_found
6: **else**
7: $description \leftarrow$ add_descriptor($description$)
8: **return** clarify($description$)
9: **end if**

Let us take a look at an example to better understand the overall process. Suppose there are two bottles on a table, b_1, a red glass bottle and b_2, a green plastic bottle. The human asks the robot for a bottle: "Give me the bottle". Thus, the initial description corresponds to $[(\texttt{type}, \texttt{bottle})]$. Since both objects fulfill this description, a new descriptor is required. Suppose we add the color information. In this case, the new description corresponds to $[(\texttt{type}, \texttt{bottle}), (\texttt{color}, \texttt{red})]$. The algorithm ends now indicating that the object is identified as b_1, the red glass bottle.

In order to add a new descriptor (attribute-value pair) two alternatives are available: directly asking the human for a new descriptor, or automatically searching a new attribute and ask the human for its value. In the latter case, we need to automatically find the best discriminant for the current list of objects being evaluated ($objectL$ in the algorithm).

Finding a discriminant: We have implemented a set of semantic categorization functions in ORO. One of them consists in looking for discriminants, i.e. descriptors that allow a maximum discrimination among a set of individuals. In the example above, considering the attributes *type, color* and *material*, ORO would return *color* and *material* as discriminants, since their values are unique for the given set of objects.

We distinguish two types of discriminants. *Complete* discriminants are those attributes (or properties) that totally discriminate the set of individuals. In other words, properties whose values can uniquely identify those individuals. However, they are not always available. First, because two or more individuals may share the same value, and second, because not all individuals may share the same properties. Thus, *partial* discriminants are those that "better" split the set of individuals in different subsets based on some criteria.

The algorithm to determine the type of discriminant available (Algorithm 2) has the following steps (to better follow it, we base its description on the ontology example illustrated in Figure 2. We search a discriminant for the following individuals: $plant_1, animal_1, animal_2$ and $animal_3$). First we obtain the direct properties for all the individuals, i.e. we do not consider all the hierarchy of properties (line 1). In the example, $plant_1$ has two superclasses (`plant` and `thing`), but we only take the most direct one (the class `plant`). Next, we compute the number of individuals per property (line 4) and the number of different values for that property (line 5). If there is more than one different value for the property (in other words, if not all individuals have the same value), then we consider that property as a potential discriminant (lines 6 and 7). Finally, we sort the list of potential properties following two criteria: the number of individual occurrences (i.e. the most individuals are covered by that property, the better) and the values occurrences (i.e. the more distinct values, the better). The best discriminant corresponds to the first element of the sorted list. In other words, the class with higher number of occurrences and more variety in it. If several properties are equal, return all of them.

In our example, the algorithm would return the class name as the partial discriminant. If we only consider the instances of the class `Animal`, it would return two properties equally discriminant: {`hasColor`,`eats`}. It should be noted that this way of proceeding does not respect the open world assumption. We believe that the robot should only reason bases on his current knowledge.

IV. Integration and Validation Tasks

In order to validate our approach, we have use two types of tasks. The first one corresponds to a daily-life situation where a human asks the robot for an object providing partial information, while the second one is focused on a child game: the Spy-Game. Figure 3 illustrates the scenario for both tasks. The relevant objects of the scenario are described through the features presented in Section II. Table I shows some of the objects indicating wether the description is manually given or automatically computed. For example,

Algorithm 2 get_discriminant($individuals$)

$P \leftarrow$ get_properties($individuals$)
$\hat{P} \leftarrow nil$
for all $p \in P$ **do**
  $n_{ind} \leftarrow$ nb_ind_with_prop(p)
  $n_{val} \leftarrow$ nb_diff_values(p)
  if $n_{val} > 1$ **then**
    $\hat{P} \leftarrow append([p, n_{ind}, n_{val}])$
  end if
end for
sort($\hat{P}$)
return $first(first(\hat{P}))$

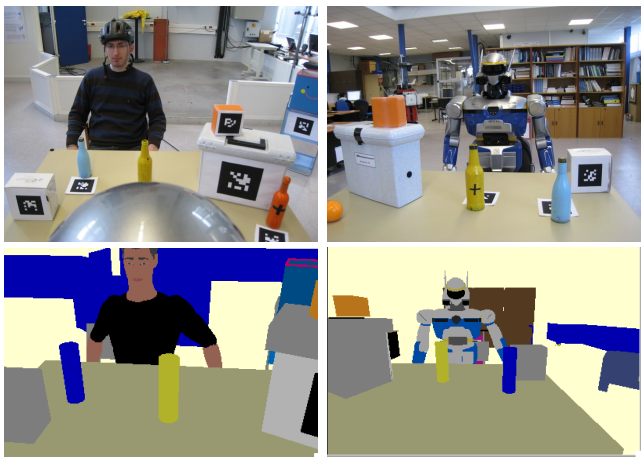

Fig. 3. Scenario for validation tasks. Top row correspond to the real environment, while the bottom row are snapshots of the modeled environment. Robot's visual perspective (left) and human's visual perspective (right).

human	Give me the ball.
robot	*(Possible objects: objectL = [purple-ball, orange-ball])*
	(Removing non visible objects for the human: objectL = [orange-ball])
	You want the orange-ball.
human	Give me the bottle.
robot	*(Possible objects: objectL = [blue-bottle, yellow-bottle, orange-bottle, red-bottle])*
	(Removing non visible objects for the human: objectL = [blue-bottle, yellow-bottle])
	(Complete discriminants: color.)
	Which one: blue or yellow?
human	Blue.
robot	*(Possible objects: objectL = [blue-bottle])*
	You want the blue-bottle.
human	Give me the box on the table.
robot	*(Possible objects: objectL = [accesskit, white-box])*
	(Removing non visible objects for the human: objectL = [accesskit, white-box])
	(Complete discriminants: none (both boxes share type, shape))
	Which one? I need more information.
human	On your left.
	(Possible objects: objectL = [white-box])
	You want the white-box.

Fig. 4. "Which one?" example. The uppermost cell includes an ambiguous situation solvable through visual perspective taking, the middle cell a feature-based solution, the lower cell a spatial perspective taking based solution.

the information provided for the object *orange-bottle* is that it belongs to the class Bottle and that its color is orange. Based on the knowledge in the common sense ontology, the robot can infer it is also a Tableware, and that its shape is cylindrical. The remaining descriptors (location, spatial and visual perspective taking) are automatically computed through geometric reasoning. In both tasks, the agents communicate by using screen and keyboard. Both tasks have been performed in simulation as well with two robotic platforms: Jido and HRP-2.

A. Integration

The whole system is integrated into both robotic platforms as components of the LAAS architecture [12]. In order to acquire and keep a coherent model of the environment, three main modules are used: the *Object Recognition Module* detects and localizes objects through markers; the *Human Detection Module* localizes and tracks the human looking orientation through motion capture cameras; and the *Robot Manager Module* provides the robot's current configuration. The *Geometric Reasoner* constantly updates its 3D environment while ORO is updated when required (either through interaction or automatically through geometric reasoning).

B. Task 1: Which one?

In this task we are considering a daily situation where the human needs an object and asks the robot to pass it over. The human query either provides complete information, i.e. based on the description the referent is identifiable, or partial information resulting in an ambiguity, i.e. more than one object matches the description. In the first case, the robot can directly hand the object since only one object matches the description. However, in cases where more than one object matches the query, the robot should try to resolve the ambiguity efficiently. Based on the algorithm presented in Section III-B, the robot tries to automatically find out a complete discriminant that will allow it to resolve the ambiguity in one single question (asking for the attribute's value). If no complete discriminant is available, it is an efficient strategy to ask the human for more specific information (an attribute-value pair) resulting in a fast referent identification. Figure 4 describes three situations where type of information provided by the human varies.

C. Task 2: Spy-Game

This game is based on the traditional children game "I Spy". The idea is to discover the object (which should be visible to everyone) or concept one of the participants is

Id	Type	Color	Shape	Location	Spatial PT		Visual PT	
					Robot	Human	Robot	Human
purple-ball	**ball**	**purple**	sphere	on $shelf_1$	back near	front far	false	false
orange-bottle	**Bottle**, Tableware	**orange**	cylinder	on big_table	front right near	front left near	true	false
red-bottle	**Bottle**, Tableware	**red**	cylinder	in $trashbin$	front right far	back left near	false	false
orange-box	**Box**, Object	**orange**	**cube**	on $accesskit$	front near	front left near	true	true

TABLE I

EXAMPLE OF OBJECTS' DESCRIPTIONS. IN BOLD, INFORMATION GIVEN AT INITIALIZATION. ALL OTHER DESCRIPTORS ARE AUTOMATICALLY COMPUTED BY THE GEOMETRIC REASONING OR INFERRED FROM THE ONTOLOGY ON-LINE.

human	It is a tableware.
robot	*(Possible objects: objectL = [blue-bottle, yellow-bottle, orange-bottle, red-bottle and cup-with-handle])*
	(Removing non visible objects for the human and the robot: objectL = [blue-bottle, yellow-bottle, cup-with- handle])
	(Discriminants: type, color.)
	Which type of object it is: bottle or cup?
human	Bottle.
robot	*(Possible objects: objectL = [blue-bottle, yellow-bottle])*
	[Discriminants: color.]
	What color the object is: blue or yellow?
human	Blue.
robot	*[Possible objects: objectL = [blue-bottle])*
	The object is the blue bottle!

Fig. 5. Spy Game example.

thinking of by asking questions such as: "Is it green? Is it a machine? Is it on your left?", etc. When playing, children exploit their knowledge about the world while categorizing and describing objects through useful discriminants.

The Spy Game (see Figure 5 for an example) starts with the human user giving a first hint that corresponds to the initial description indicated in the algorithm introduced in Section III-B. The robot tries to figure out the object automatically searching for discriminants (complete or partial) and querying the human for their values until either discovering the object or giving up (no object matches the description).

V. CONCLUSIONS AND FUTURE WORK

Grounding the referent is essential for a robot to interact with humans. Humans constantly generate and resolve ambiguities, and therefore, they expect that robots will be able to do so as well. Thus, we believe that it is important to include various clarification strategies helping the robot to better understand its human partner. However, to interact successfully with a human partner, the knowledge and reasoning processes available to the robot are critical. In this work we have presented different sources of information to feed the robot's knowledge, as well as an ontology to store, manage and reason on it. We have introduced an algorithm that allows the robot to detect and resolve ambiguous situations arising in natural interaction. Two application tasks have been described: Spy Game and object identification. The validation tasks were successfully performed both in simulation and on two different robot platforms (HRP-2 and Jido).

Although we have performed a step forward in solving ambiguities, there is still a lot of work to do. The most immediate one is to integrate the use and recognition of deictic gestures, such as pointing and showing (humans use these types of gestures) as another source of information for clarification. We also plan to include comparative reasoning among a set of objects to identify properties such as: the bigger one, the nearest one, etc. Finally, we also plan to extend the robot's knowledge by learning new concepts based on the descriptions obtained when failure occurs.

REFERENCES

[1] H. Moll and M. Tomasello, "12- and 18-month-old infants follow gaze to spaces behind barriers," *Developmental Science*, vol. 7, no. 1, pp. F1–F9, 2004.

[2] ——, "Level 1 perspective-taking at 24 months of age," *British Journal of Developmental Psychology*, vol. 24, 2006.

[3] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. E. Mintz, and A. C. Schultz, "Enabling effective human–robot interaction using perspective-taking in robots," *IEEE Transactions on systems, man and cybernetics - Part A: Systems and Humans*, vol. 35, no. 4, pp. 460–470, 2005.

[4] J. G. Trafton, A. C. Schultz, M. Bugajska, and F. Mintz, "Perspective-taking with robots: Experiments and models," in *Int Workshop on Robots and Human Interactive Communication*, 2005, pp. 580–584.

[5] M. Berlin, J. Gray, A. L. Thomaz, and C. Breazeal, "Perspective taking: An organizing principle for learning in human-robot interaction," in *In Proceedings of AAAI*, 2006.

[6] M. Johnson and Y. Demiris, "Perceptual perspective taking and action recognition," *International Journal of Advanced Robotic Systems*, vol. 4, no. 4, pp. 301–308, 2005.

[7] L. F. Marin-Urias, E. A. Sisbot, and R. Alami, "Geometric tools for perspective taking for human-robot interaction," in *Mexican International Conference on Artificial Intelligence*, 2008, pp. 243–249.

[8] R. Ros, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Solving ambiguities with perspective taking," in *In Int. Conf. on Human-Robot Interaction*, 2010, pp. 181–182.

[9] B. Tversky, P. Lee, and S. Mainwaring, "Why do speakers mix perspectives?" *Spatial Cognition and Computation*, vol. 1, 1999.

[10] H. A. Taylor, S. J. Naylor, R. R. Faust, and P. J. Holcomb, "Could you hand me those keys on the right? disentangling spatial reference frames using different methodologies," *Spatial Cognition and Computation*, vol. 1, no. 4, pp. 381–397, 1999.

[11] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, and M. Beetz, "Oro, a knowledge management module for cognitive architectures in robotics," in *IROS*, 2010, to appear.

[12] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *Int. Journal of Robotics Research*, vol. 17, pp. 315–337, 1998.